

飞浪脚本零起点入门系列(六)

MAXScript 的条件选择语句和找面数为 0 的物体

作者：飞浪

声明:本教程为 **CG++** 原创,转载请注明出处,谢谢合作:)

本节关键词: `if...then...[else]` `if...do` `case...of`

前面讲过数组,讲过 for 循环,程序里面还有一个重要的组成部分,那就是今天要讲的条件选择语句。大家知道程序是非常严格的,对就是对,错就是错,不像哲学那么似是而非,那么如何通过程序语句判断对错呢?首先登场的是程序语句的一个重量级语句:

If Expression (If 表达式)

If Expression 在线参考手册:

http://www.cgplusplus.com/online-reference/maxscript-reference/source/if_expression.htm

语法形式有两种类型:

`if <expr> then <expr> [else <expr>]` 或者 `if <expr> do <expr>`

<expr>是表达式, if 后面的<expr>必须返回布尔常量 true 或者 false, 即对或者错。

then, else, do 后面的<expr>可以是任何其他 maxscript 表达式。中括号表示可选择。

一个简单的例子:

```
a = 1
```

```
b = 2
```

```
if a < b then print "a 小于 b"
```

 --这句执行是这样的: 首先执行 if 后面的表达式: a<b 返回结果对(true), 是正确的就继续执行后面的 print "a 大于 b"就输出了字符串"a 大于 b"。

如果把它改一下:

```
if a > b then print "a 大于 b"
```

 --首先判断 a > b 返回错(false), 那么 then 后面的语句就不继续执行了。再来看:

```
if a > b then print "a 大于 b" else print "a 小于 b"
```

 --跟上面一样, 不执行 then 后面的语句, 但是程序自动找到了后面的 else, 而执行 else 后面的语句。这句翻译过来是: 如果 a>b, 那么执行 print "a 大于 b"否则执行 print "a 小于 b"。这样就可以通过判断一个语句的 true 或者 false 来执行相关操作了。

下面讲一个 maxscript 计算面数的函数:

getPolygonCount <node> --计算节点<node>的面数和顶点数, 返回一个数组。

例如, 选择一个茶壶, 执行 `getPolygonCount $` 即计算此茶壶的面数和点数, 返回结果是:

```
#(1024, 530)
```

这个数组里面, 第一个值表示此节点的面数, 第二个值表示节点的顶点数, 通过前面讲过的

数组我们知道，调取此物体的面数可以这样做：

```
a = getPolygonCount $ --获取选择节点的面数和点数
```

```
a[1] --返回物体的面数
```

我们看到有些脚本里面有选择面数为 0 的物体，那么这个脚本是怎么写的呢？

例子：

```
obj = #()
for i in geometry do
(
    a = getPolygonCount i
    if a[1] == 0 then append obj i
)
if obj.count !=0 then select obj else print "没有找到面数为 0 的物体！"
```

逐行解释：

obj = #() --先定义一个空数组，用来收集面数为 0 的物体

```
for i in geometry do
```

```
(    --在所有的几何体中逐个寻找
```

```
    a = getPolygonCount i --获取节点 i 的面数和点数
```

```
    if a[1] == 0 then append obj i --如果 i 的面数为 0,那么就把它加入到数组 obj 中
```

```
)
```

```
if obj.count !=0 then select obj else print "没有找到面数为 0 的物体！"
```

--如果收集的物体个数不为 0 就选择收集的物体，否则，输出“没有找到面数为 0 的物体！”

如果你选择一个灯光，然后用 getPolygonCount \$也可以返回此灯光的面数和顶点数，不过结果是#(0,0)，即没有面数，嘿嘿！

再回来看最开始那个例子，可以改成这样的：

```
a = 1
```

```
b = 2
```

```
if a < b do print "a 小于 b" --把 then 换成了 do，效果一样的，那么 then 和 do 有什么不同呢？
```

if...then 和 if...do 不同点是，if...then 后面接表达式后还可以接 else 执行其他语句，而 do 后面只能接一个表达式，别无他的选择了。所以在 maxscript 的编写中，建议扔掉 if...do 语句而用 if...then，这样在程序的修改上有大大的好处。比如你写个语句：

```
if a < b then print "a<b" else () --else 后面接一个空语句()，虽然不执行任何操作，但是如果后期想修改，直接在()里面添加内容即可，如果用 if...do，后面就不能再加其他的选择了，你又要改成 if...then...else。
```

Case Expression (Case 表达式)

Case Expression 在线手册 :

http://www.cgplusplus.com/online-reference/maxscript-reference/source/case_expression.htm

Case 表达式语法形式:

```
case [ <expr> ] of ( <cases> )
```

[<expr>]是可选表达式, <cases>是一系列可能会执行的内容

<cases>语法是:

```
<factor>: <expr>
```

```
default: <expr>
```

例如, 选择一个灯光, 执行下列语句:

```
case $.enabled of
(
    true: print "打开的"
    false: print "关闭的"
)
```

逐行解释:

```
case $.enabled of
```

(--\$.enabled 是灯光打开或关闭的参数, 可以读取, 也可以设置, 如果你单独执行一下\$.enabled, 就可以看到选择灯光的打开状态, 是 true 或者 false

```
    true: print "打开的"
```

--true 和 false 都是<factor>, 即可供选择的因子

```
    false: print "关闭的"
```

)

此程序的执行是这样的, 首先执行\$.enabled 得到此灯光的开关状态, 比如说关闭(false), 然后到下面的因子中逐行往下寻找 false 的因子, 当遇到**第一个** false 时, 就执行此 false 后面的表达式然后结束此 case 语句。但是如果一直不到 false 怎么办呢? 那么结果就是 undefined, 为避免此类情况发生, 我们可以再最后面加一个<factor>即 default, 如果找不到任何正确的<factor>, 就可以执行默认的表达式。例如:

```
a = 1;b = 1;
```

```
case of
```

```
(
```

```
    (a > b): print "a 大于 b"
```

```
    (a < b): print "a 小于 b"
```

```
    default: print "其他"
```

```
)
```

这个例子里面, case 后面直接用 of 而省略了表达式, 如果这样写, 那么下面的每个<factor>就要返回布尔常量, 如(a > b)返回 false, (a < b)也返回了 false, 这样这两个后面的程序都不会执行, 到最后 default, 即执行默认的程序。

下面请大家自己动手, 写一些简单常用的程序, 请用到 if 语句:

- 1、找所有没有上材质的物体 (提示: 没材质的判断方法为: \$.material == undefined)
- 2、收集场景中所有的 standard 材质 (提示: 用 classof 判断材质类型, 见第五节)
- 3、调场景中所有 VRayLight 的细分到 20
- 4、把场景中所有类型为 Teapot 的线框颜色改成红色
- 5、判断一个 box 的长, 大于、等于或小于 100 的情况, 然后输出相应字符串 (请用 case 语句)

这些是为了使大家更清楚的明白 if 和 case 语句的原理, 希望大家踊跃的试试。你做了哪一道题, 请在下面写上题目内容以及你的答案。请大家继续支持此系列教程。

本节原帖地址: <http://www.cgplusplus.com/bbs/viewthread.php?tid=174>

本节结束。